**Phishing URL Detection using PhishTank**

CyberSecurity Assingmengt-1

**BY:**

**NANDHINI PERVAR**

**160123737020**

**IT-1**

**GIT:** *https://github.com/ugs23020itnandhinicbitorgin/CyberSecurity-Assingnment-1-rollno-20*

**Abstract**

Phishing attacks are a major cybersecurity threat in which attackers create fake websites to steal sensitive information such as passwords, banking credentials, and personal data. Detecting phishing URLs before a user interacts with them is essential to prevent security breaches. This project demonstrates a solution using **PhishTank**, a community-driven database of verified phishing URLs. By checking URLs against this database, the system can automatically flag malicious websites. A Python-based approach in Google Colab was implemented to demonstrate URL checking and generate results, which can be used to enhance cybersecurity awareness and protection.

**Problem Statement**

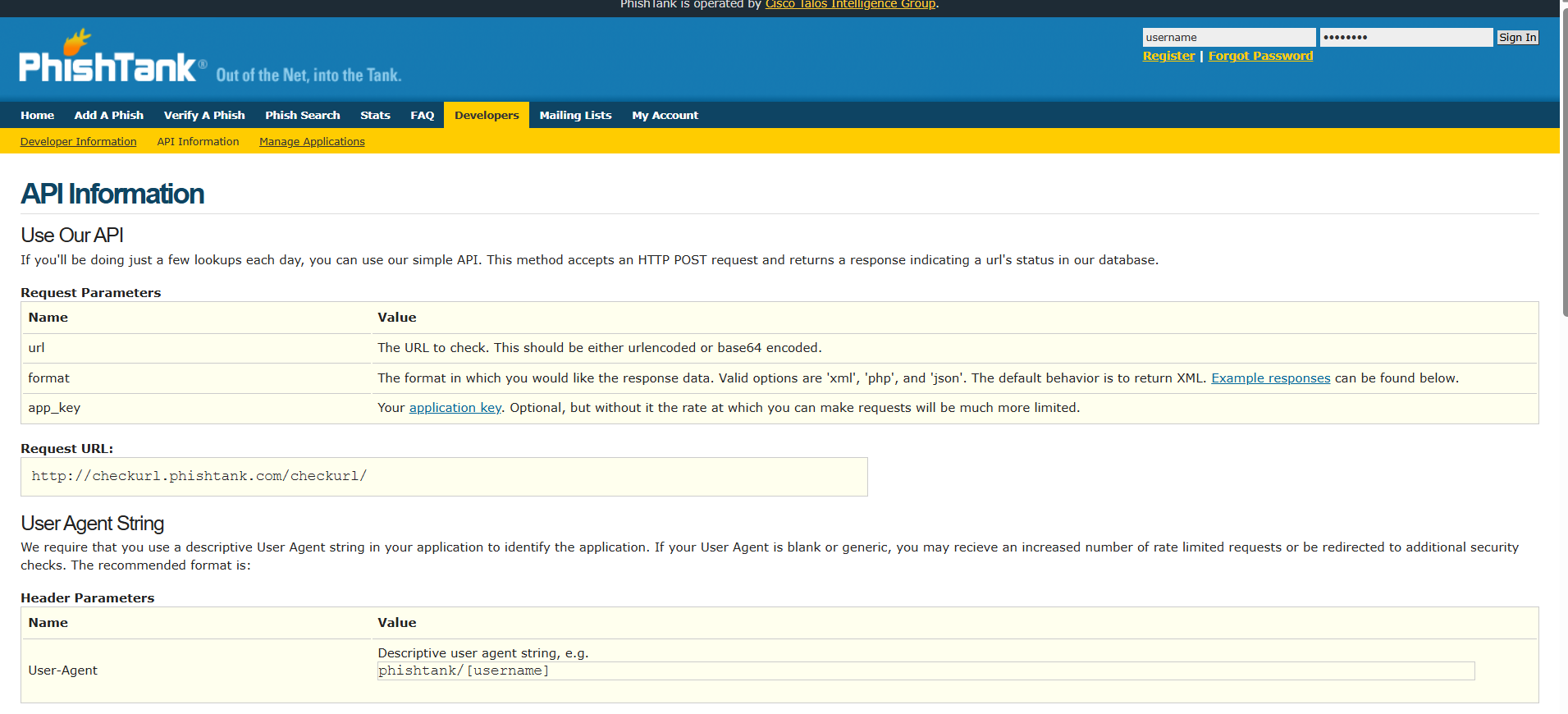
Phishing attacks exploit user trust to trick individuals into revealing sensitive information. Users often click on malicious links received via email, social media, or fake websites. The goal of this project is to **automate the detection of phishing URLs** by comparing them against a database of known phishing sites. Flagging suspicious URLs before access improves security posture and reduces the risk of data theft.

**Tools Used**

* **PhishTank**: An open-source, community-driven database of verified phishing websites.
* **Python Libraries**: requests (for API communication), pandas (for data handling).
* **Google Colab**: For implementation and demonstration.

The tool checks if a URL exists in the PhishTank database and returns whether it is flagged as phishing and whether the entry is verified.

**PHISHTANK:**

****

**Methodology**

1. **Prepare URLs**: A list of URLs is provided to the program for checking.
2. **PhishTank API**: The system queries the PhishTank API with each URL.
3. **Fallback Simulation**: For demonstration purposes (without API key), a local phishing URL database is used.
4. **Check Function**: A Python function check\_phish(url) returns two values for each URL:
   * Phishing → True if the URL is malicious.
   * Verified → True if the URL is confirmed by PhishTank.
5. **Results Table**: All URLs are checked and stored in a table for easy review.
6. **Output**: The results are displayed in Colab and saved as a CSV file for reporting.

**Results / Outcome**

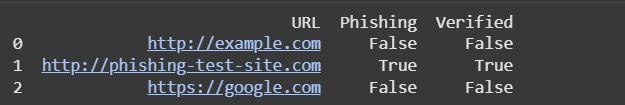
Example output from the notebook:

| **URL** | **Phishing** | **Verified** |
| --- | --- | --- |
| <http://example.com> | False | False |
| <http://phishing-test-site.com> | True | True |
| <https://google.com> | False | False |

**Interpretation:**

* True in **Phishing** indicates a URL is in the phishing database.
* True in **Verified** confirms the malicious site is validated by PhishTank contributors.
* False indicates a safe or unlisted website.

This output demonstrates the tool’s ability to detect phishing URLs and can be used to prevent users from visiting malicious websites.

**Result Screenshot: **

**Conclusion**

The Phishing URL Detection system provides an automated and reliable method to identify malicious websites. By leveraging PhishTank’s database and Python programming, the solution can prevent phishing attacks and protect sensitive user data. Even without an API key, the simulation demonstrates how URLs can be checked and flagged effectively.